

**IN THE CLAIMS**

2. (Currently Amended) A method for monitoring an event within a hardware description language (HDL) model, said method comprising:

generating an HDL simulation model;

specifying an event within said HDL simulation model as an output port of an instrumentation entity;

uniquely naming said event within an event translation table;

delivering an event monitor request within said HDL simulation model; and

responsive to said event monitor request, retrieving said event from said instrumentation entity in response to said event monitor request, wherein said retrieving said event comprises retrieving said unique name from said event translation table.

3. (Original) The method of claim 2, further comprising utilizing input port map comments to generate a connection to said event from within said HDL simulation model.

4. (Currently Amended) The method of claim 2, further comprising utilizing entity declarations within an HDL source code file to generate and uniquely name said event.

5. (Canceled)

6. (Currently Amended) The method of claim ~~[[5]]~~ 2, further comprising constructing said event translation table from said entity declaration comments during a model build process.

7. (Currently Amended) A system for monitoring an event within a hardware description language (HDL) model, said system comprising:

a simulator that simulates said HDL model;

a simulator controller that delivers an event monitor request within said HDL simulation model;

an instrumentation entity that generates an event within said HDL model, wherein said instrumentation entity is produced by an HDL source code file comprising input port map comments that generate a connection to said event from within said HDL simulation model, wherein said HDL source code file further comprises entity declaration comments that generate and uniquely name said event; and

an application program interface (API) that retrieves said event from said instrumentation entity in response to said event monitor request from said simulator controller; and

an event translation table that associates a unique name with said event, wherein said API retrieves said unique name from said event translation table in response to said event monitor request from said simulation controller.

8. (Original) The system of claim 7, wherein said simulator controller is a run time executive that calls an application program from said API.

9. (Canceled)

10. (Canceled)

11. (Canceled)

12. (Currently Amended) The system of claim ~~[[11]]~~ 7, further comprising an instrumentation load tool for constructing said event translation table from said entity declaration comments during a model build process.